1.2.a) Refer to the documentation, what is the functionality of the tol parameter in the Perceptron class? (2 marks)

Answer:

Tol stands for tolerance level for improvement in loss function. When improvement in loss is smaller than the tol value, it stops the epoch.

1.2.b) If we set max_iter=5000 and tol=1e-3 (the rest as default), does this guarantee that the algorithm will pass over the training data 5000 times? If not, ensure that the algorithm will pass over the training data 5000 times? (2 marks)

Answer:

It does not guarantee that the algorithm will pass over the training data 5000 times. Tuning the tol parameters to a smaller value can make sure that the algorithm will pass over the training data 5000 times.

1.2.c) How can we set the weights of the model to a certain value? (2 marks)

Answer:

We can set class_weight when initializing the perceptron class. It is used to adjust importances of different clases. Class_weight consists of a dictionary to set the pair value of the initial weight and feature. We can also access and modify coef_ attributes to set weights of the model.

1.2.d) How close is the performance (through confusion matrix) of your NumPy implementation in comparison to the existing modules in the scikit-learn library?

Answer:

Our Confusion Matrix:

```
[[8. 0.]
 [3. 9.]]
```

Through Scikit-learn library:

```
[[ 8  0]
 [ 2 10]]
```

Based on our observation, our matrix has one more in false negative classification and one less in true positive classification than the results from scikit-learn library.

2.1.a) When we input a singular matrix, the function linalg.inv often returns an error message. In your fit_LinRegr(X_train, y_train) implementation, is your input to the function linalg.inv a singular matrix? Explain why. (2 marks)

Answer:

Yes, the matrix is a singular matrix. The rows are linearly dependant, where the second row is 2 times the the first, the third row is 3 times the first row, and the fourth row is 4 times the first. The matrix only has one independent row, so the rank of the input matrix is 1.

2.1.b) As you are using linalg.inv for matrix inversion, report the output message when running the function subtestFn(). We note that inputting a singular matrix to linalg.inv sometimes does not yield an error due to numerical issue. (1 marks)

Answer:

The output is “ERROR”.

2.1.c) Replace the function linalg.inv with linalg.pinv, you should get the model’s weight and the “NO ERROR” message after running the function subtestFn(). Explain the difference between linalg.inv and linalg.pinv, and report the model’s weight. (2 marks)

Answer:

Linalg.inv computes inverse of a square matrix, which is not singular. Whereas, linalg.pinv computes the pseudo-inverse of a matrix, which can be singular.

The model’s weight is `weights:  [1.04360964e-14 2.00000000e-01 4.00000000e-01]`